Evaluating spatial design techniques for solving land-use allocation problems.

Aerts, J.C.J.H.; van Herwijnen, M.; Janssen, R.; Stewart, T.J.

published in
Journal of Environmental Planning and Management
2005

DOI (link to publisher)
10.1080/0964056042000308184

document version
Publisher's PDF, also known as Version of record

Journal of Environmental Planning and Management
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713429786

Evaluating Spatial Design Techniques for Solving Land-use Allocation Problems

Jeroen Aerts[a]; Marjan Van Herwijnen[a]; Ron Janssen[a]; Theodor Stewart[b]
[a] Institute for Environmental Studies (IVM), Vrije Universiteit Amsterdam, Amsterdam, The Netherlands [b] Department of Statistical Sciences, University of Cape Town, Rondebosch, South Africa

Online publication date: 03 August 2010

To cite this Article Aerts, Jeroen , Van Herwijnen, Marjan , Janssen, Ron and Stewart, Theodor(2005) 'Evaluating Spatial Design Techniques for Solving Land-use Allocation Problems', Journal of Environmental Planning and Management, 48: 1, 121 — 142

To link to this Article: DOI: 10.1080/0964056042000308184

URL: http://dx.doi.org/10.1080/0964056042000308184

Journal of Environmental Planning and Management,
Vol. 48, No. 1, 121–142, January 2005

Evaluating Spatial Design Techniques for Solving Land-use Allocation Problems

JEROEN C. J. H. AERTS*, MARJAN VAN HERWIJNEN*, RON JANSSEN* & THEODOR J. STEWART**

*Institute for Environmental Studies (IVM), Vrije Universiteit Amsterdam, Amsterdam, the Netherlands

**Department of Statistical Sciences, University of Cape Town, Rondebosch, South Africa

(Received September 2003; revised January 2004)

ABSTRACT *This study examines the use of spatial optimization techniques for multi-site land-use allocation problems (MLUA). 'Multi-site' refers to the problem of allocating more than one land-use type in an area, which are difficult problems as they involve multiple stakeholders with conflicting goals and objectives. Spatial optimization methods consist of (1) an optimization model and (2) an algorithm to solve the model. This study demonstrates a goal-programming model to solve the MLUA problem. The model is solved using both simulated annealing and genetic algorithms. Special attention has been given to introduce a spatial compactness objective in the model. It is shown that the compactness objectives in the optimization model generate compact patches of the same land use for using both the simulated annealing procedure and the genetic algorithm. In addition, it appears that using the proper settings of the compactness objectives, connectivity between patches of land use is promoted. The method is tested for a fictive study and then demonstrated for a real case study, both measuring 20 × 20 cells. The genetic algorithm generally performs better than simulated annealing in terms of solution time and achieving compactness.*

Introduction

Land-use allocation problems deal with the planning of new land uses to an area or re-distribution of existing land uses within an area. These problems are often complex as they involve multiple stakeholders with conflicting goals and objectives (e.g. White & Engelen, 1997). Therefore, much attention has been paid to solving land-use allocation problems with multi criteria decision-making techniques (MCDM). Recent research focuses on combining MCDM with a geographic information system (GIS). This appears to be a powerful combination, since land-use allocation problems both involve multiple objectives and criteria as well as geographically dependent spatial attributes (Cova, 1999; Cova & Church, 2000a, 2000b; Aerts, 2002; Aerts & Heuvelink, 2002).

Correspondence Address: Jeroen C. J. H. Aerts, Institute for Environmental Studies (IVM), Vrije Universiteit Amsterdam, De Boelelaan 1115, 1081 HV Amsterdam, The Netherlands.
Email: Jeroen.Aerts@ivm.vu.nl

ISSN 0964-0568 Print/1360-0559 Online/05/010121-22
DOI: 10.1080/0964056042000308184

A type of a land-use allocation problem is multi-site land-use allocation (MLUA). This refers to the problem of allocating more than one land-use type in an area. A difficult aspect in MLUA problems are conflicting interests in allocating these land uses. Think of, for instance, developing a new piece of land with three land-use types: agricultural land, industry and housing. The area may be suitable for developing agricultural land, but can be developed as a residential area as well. The latter would be somewhat more expensive, but more environmentally friendly since less groundwater will be contaminated. Hence, a planning question to be solved by optimization techniques would be to optimally allocate agricultural and urban areas while preserving environmental quality and agricultural production. This depends, for example, on the different land requirements, such as development costs, environmental damage, etc.

All these aspects are addressed in an optimization model, which is a well-known MCDM technique (Malczewski, 1999; Church, 2002). In this study, the use of spatial, GIS-based, optimization techniques will be examined for solving problems. More specifically, it is proposed to begin exploring whether a so-called goal programming-based approach is suitable for solving an MLUA problem (e.g. Stewart, 1991; Ridgley *et al*., 1997; Stewart *et al*., 2002). It is believed, in this respect, that goal-programming techniques may accommodate a decision maker in the area of land-use planning as they feel comfortable with defining goals and from there develop land-use plans.

A crucial element in the development of a spatial optimization model for MLUA problems is to introduce spatial compactness objectives in the model. Spatial compactness objectives are used to address the problem of allocating the same land use not only at lowest cost but also at maximum compactness (e.g. Wright *et al*., 1983; Williams & Revelle, 1998; Cova & Church, 2000a; Aerts, 2002; Aerts *et al*., 2003). Compact pieces of land use are often seen as an indicator for environmental quality; the more compact the land the greater potential for species. Moreover, a challenge lies in developing compactness objectives that promote contiguity or connectivity of patches of the same land use in an area. Contiguity is often required in ecological studies. For example, think of finding an ecological corridor that connects two pieces of nature reserves.

From the former, the following objectives are arrived at for this study:

- To develop a goal-programming model based on a reference point approach that can solve an MLUA problem. The problem will be solved by using both a simulated annealing procedure and a genetic algorithm.
- To develop different spatial compactness objectives in order to provide a decision maker with different land-use plans.
- To test the model on both its efficacy to encourage spatial compactness as well as on connectivity.
- To apply the model for a real case study to evaluate the performance of the simulated annealing approach against the genetic algorithm approach.

The model and two algorithms are first tested on a fictive study area. For this area, the land-use cost distributions over the area are fairly simple, which supports the interpretation of the compactness achievements of the model results. Next, the model will be applied to a real case study in the Jisperveld area in The Netherlands.

Setting up a MLUA Model

The Basic Optimization Model

An MLUA problem can be approached as an optimization problem, which is formulated as a cost function for either minimizing or maximizing (there is no real difference). For a land-use allocation problem, a basic optimization model can be formulated as follows. Consider a rectangular area in which different land uses need to be allocated. First, the area is divided into a grid with N rows and M columns. Let there be K different land-use types. A binary variable x_{ijk} is introduced which equals 1 when land use k is assigned to cell (i,j) and equals 0 otherwise. Furthermore, development costs (C_{ijpk}) are involved for each land-use type k in cell (i,j). These costs vary depending on the location which varies according to specific cost attributes p (for p = 1, …, P) such as soil type, construction costs and management costs.

The objective is to minimize costs associated with allocating land uses k (for k = 1, …, K) to a map **u**. Accordingly, the basic optimization model may be written as follows:

Minimize

$$f_p(u) = \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j=1}^{M} C_{ijpk}x_{ijk} \tag{1}$$

$$\forall \quad p = 1, \mathrm{K}, P$$

Subject to

$$\sum_{k=1}^{K} x_{ijk} = 1 \tag{2}$$

$$\forall\, i = 1, \mathrm{K}, N, \quad j = 1, \mathrm{K}, M \quad x_{ijk} \in \{0, 1\}$$

$$L_k \le A_k \le U_k \tag{3}$$

where

$$\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ijk} = A_k \quad \forall\, k = 1, \mathrm{K}, K \tag{4}$$

and

$$\sum_{k=1}^{K} A_k = N \cdot M \tag{5}$$

Equation 2 specifies that one and only one land use must be assigned to each cell. Because decision variable x_{ijk} must be either 0 or 1, the model is defined as an integer programme (IP). Equations 3, 4 and 5 bound the number of cells A_k allocated to a certain land-use type k between an upper and lower limit, expressed as L_k and U_k , respectively.

If the model is linear, such as the simple model described above, the model can be solved with a linear solving algorithm. However, if the model is not linear it will be necessary to use a so-called heuristic algorithm.

A More Advanced Optimization Model

The objective function described in the previous section can be expanded with a second, and if needed with more spatial objectives. This second objective refers to spatial attributes as compactness or contiguity of land use of equal type. Solving these models can be a complex task, since MLUA problems may be classified as combinatorial optimization problems, which are characterized by a very large number of possible solutions (Diamond & Wright, 1989; Greenberg, 2002). The difference between contiguity and compactness should be noted in this respect (Figure 1). Contiguity requires all cells of equal land use to be connected (Figure 1, middle). Compactness merely encourages cells of equal land use to be allocated next to one another, but this may result in divided patches (Figure 1, right). Due to limitations of space, the study will be restricted to the compactness objective.

Spatial compactness objectives are, for instance, found in forestry research harvest schedules, which deal with strict adjacency constraints (e.g. Jones *et al*., 1991; Lockwood & Moore, 1993; Murray & Church, 1995). Some studies in geographic information science have approached spatial compactness in optimization modelling by rewarding cases where neighbouring cells have equal land use (Aerts & Heuvelink, 2002; Aerts *et al*., 2003).

As outlined in the Introduction, the ultimate goal of developing a combined GIS-MCDM approach is to develop techniques that are suitable for implementation in a computer software for spatial planning, also referred to as an SDSS (Spatial Decision Support System). Therefore, in order to accommodate flexibility for the user, it is proposed to develop a mix of three spatial compactness measures, from which a user can choose or can use combinations. The spatial compactness objectives merely address commonly used compactness characteristics of clusters of the same land use. These are size, perimeter and area of a cluster (e.g. Diamond & Wright, 1989; Cova, 1999). This study is restricted to the following spatial compactness

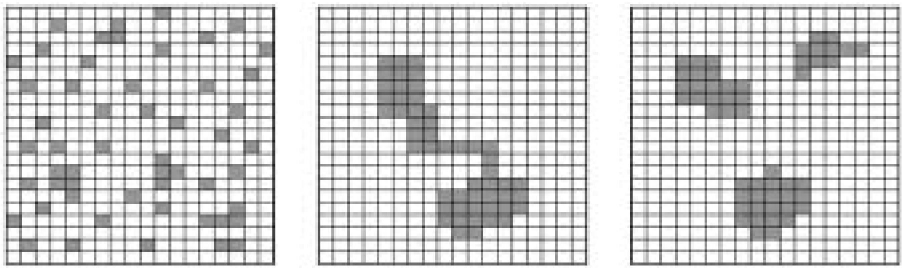

Figure 1. Area with single land use (light grey) covering 52 cells. These cells are randomly placed before optimization (left). The cells are allocated by optimizing contiguity (middle) and compactness (right).

objectives, which are determined by the extent in which the different land uses are connected or fragmented across the region:

- Spatial objective A: minimizing the number of clusters per land-use type. Less clusters of a certain land-use type points to higher compactness and less fragmentation. Hence, the ideal compactness objective value would be 1.
- Spatial objective B: maximizing the largest cluster relatively to the other clusters identified under spatial objective 1. It is preferable to have at least one large compact cluster, rather than all clusters being compact but small. The ideal compactness objective value would be 1 again.
- Spatial objective C: minimizing the perimeter of a cluster. In order to transform this measure size and scale independent, the perimeter is divided by the square root of the cluster area. The ideal compactness value would be 4, since the perimeter of a minimum cluster area -one cell- is 4. So 4 divided by the square root of 1 makes 4.

The calculation of the spatial objectives is illustrated in Figure 2. In this Figure, the value for spatial objective A is 4, because 4 clusters of the same land use can be identified. The value for spatial objective B is 0.25, which can be calculated by dividing 1 (for identifying 1 largest cluster) by 4 (total number of clusters). The value for spatial objective C is calculated using the following equation (6):

$$\sum_{r=1}^{R} \frac{H_{kr}}{\sqrt{L_{kr}}} \tag{6}$$

where H_{kr} stands for the perimeter of an identified cluster r for land use k and L_{kr} represents the area for each identified cluster r per land use k. The values for the perimeters for clusters a, b, c and d in Figure 2 are 20, 10, 18, and 22 respectively. The values for the area of all clusters are 19, 6, 5 and 25. Hence, by applying Equation 6, the value for spatial objective C becomes 70 / 14.04 = 4.98. Note, that cells within *one* cluster may connect diagonally (Figure 2).

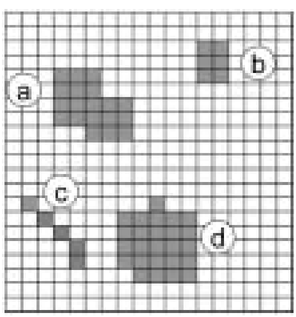

Figure. 2. Illustration of spatial compactness objectives for four clusters labelled a, b, c and d, in an area. The values for the perimeters for clusters a, b, c and d are 20, 10, 18, and 22 respectively. The values for the area are 19, 6, 5 and 25 for clusters a, b, c and d, respectively.

Goal-programming Model

Constraints and Criteria

The MLUA problem formulated above is clearly a multi-objective problem, where costs and compactness objectives have to be traded off against each other. Some research has been shown to apply a weighted sum approach where each objective is multiplied by a weight indicating its priority. Next, the result of each multiplication is added (e.g. Aerts & Heuvelink, 2002; Aerts *et al.*, 2003). In a situation where decision makers know their goals but have difficulties with valuing or weighting the relevant attributes involved, goal programming is a commonly known technique to aid decision makers with their task. A generalized goal-programming approach (reference point approach) was chosen for this study (Wierzbicki, 1999; Stewart *et al.*, 2002). Generally, in a goal-programming approach, a decision maker sets *goals* for each *objective*, and the optimization model seeks a solution that minimizes the deviation to those preset goals. A commonly used goal in land-use planning is 'total cost' of a land-use plan. An optimization model for such a problem then seeks to find a land-use plan that costs about the value set by the user of the model.

In this paper some goals (sometimes referred to as 'reference point') may be defined, say γ_p for all cost attributes related goals and λ_{kq} for all spatial objectives. The model should find a land-use map **u** for which:

$$f_p(u) \leq \gamma_p \tag{7}$$

$$s_{kq}(u) \leq \lambda_{kq} \tag{8}$$

Where $f_p\ (u)$ is the total value for all cost attributes $p\ (p = 1, \ldots, P)$ and $s_{kq}\ (x)$ the total of spatial measures $q\ (q = 1,\ldots,Q)$, which in this case is set to 3 because three compactness formulations will be included in the model.

The reference point idea of Wierzbicki (1999) uses a 'scalarizing' function, which measures under-achievement relative to the goals, but placing the greatest weight on the least well-satisfied goal. Another commonly used scalarizing function can be found in the Tschebycheff approach (Steuer, 1986) where the goal is to minimize the maximum deviation relative to the goals defined.

Here, a somewhat different scalarizing approach is used based on suggestions by Stewart (1991), which minimizes the sum of deviations but then relative to an ideal value. This approach can be defined as follows:

Minimize:

$$\sum_{p=1}^{P}\left[\frac{\int_p(u) - I_p}{\gamma_p - I_p}\right]^{\rho} + \sum_{k=1}^{K}\sum_{q=1}^{Q}\left[\frac{s_{kq}(u) - I_{kq}}{\lambda_{kq} - I_{kq}}\right]^{\rho} \tag{9}$$

Subject to: Equations (2), (3), (4) and (5).

In Equation 9, I_P is the best possible ideal value for each objective if optimized on its own, and ρ is a suitably large power. A value of $\rho = 4$ has been found to yield good results (Stewart, 1991). Advantages of this approach are (1) to avoid the use of preferences or weights, which are often difficult to interpret by the users, and (2) the function is scale free, which rules out the need for finding the worst performance levels to provide a normalized scaling.

Solving the Goal-programming Model

Heuristic Algorithms

Since both a MLUA problem can be classified as a combinatorial optimization problem, and the goal-programming model as described in the third section is non-linear in its spatial objective formulations, it is necessary to find a heuristic optimization algorithm for solving the model. Various researchers have developed linear MLUA models solved with a linear programming (LP) algorithm, but have run up against a limitation in the size of the spatial area that could be optimized (Cova, 1999; Cova & Church, 2000a, 2000b; Aerts, 2002). Heuristic approaches, however, are robust, fast and capable of solving large combinatorial problems, but they do not guarantee the optimal solution. Applications of such algorithms for MLUA problems are simulated annealing, greedy growing algorithms, genetic algorithms and Tabu search (Lockwood & Moore, 1993; Murray & Church, 1995; Brookes, 1997; Boston & Bettinger, 1999; Aerts, 2002; Aerts & Heuvelink, 2002).

The focus here is on using both simulated annealing (SA) and genetic algorithms (GA) to solve the optimization model described above. Examples of studies that use simulated annealing for spatial optimization, are research in the area of image enhancement (Sundermann, 1995), ecological applications (Church *et al.*, 1996) and applications for large harvest schedule problems in forestry research (e.g. Lockwood & Moore, 1993; Boston & Bettinger, 1999). A genetic algorithm (GA), and more generally an evolutionary algorithm mimics natural evolution processes in order to solve complex computational problems. Many studies have used GA to solve similar multi-objective problems (see e.g. Fonseca & Fleming, 1995; Jaszkiewicz, 2002).

Simulated Annealing (SA)

Kirkpatrick *et al.* (1983) introduced the concept of annealing in combinatorial optimization. This concept is explained in Figure 3. The initial situation is the current land-use map **u** for the area. The associated costs are denoted by $f(0)$. Note that costs refer to the value of Equation (9), which expresses the sum of deviations to the ideal values of all objectives. Following the flow diagram in Figure 3, the land use of a randomly chosen cell is now swapped into another randomly chosen land use. This yields a new situation, with new costs $f(1)$. Whether the change from state 0 to state 1 is accepted depends on the difference in costs $f(1)$-$f(0)$. Once this is decided, the swapping procedure is repeated, and if it is decided whether the change is accepted, a new swap should be generated, and so on. Whenever the costs $f(1)$ are smaller than the costs $f(0)$, the cell change is accepted. When $f(1) > f(0)$, costs are accepted with a certain probability following the Metropolis criterion Equation (11) (e.g. Aarts & Korst, 1989). This is achieved by comparing the value of the Metropolis criterion with a random number drawn from a uniform [0,1) distribution (Figure 3).

$$P(accept\ change) = \exp\left(\frac{f(0) - f(1)}{s_0}\right) \tag{11}$$

where s_0 is a *control* or *freezing* parameter.

A crucial element of the procedure is the gradual decrease of the freezing parameter s_i (Laarhoven, 1987). Usually, this is done by using a constant multiplication factor:

$$s_{i+1} = r \cdot s_i \tag{12}$$

where $0 < r < 1$. This effectively means that jumping to higher energy (read: costs) becomes less and less likely towards the end of the iteration procedure (Sundermann, 1995; Levine, 2002).

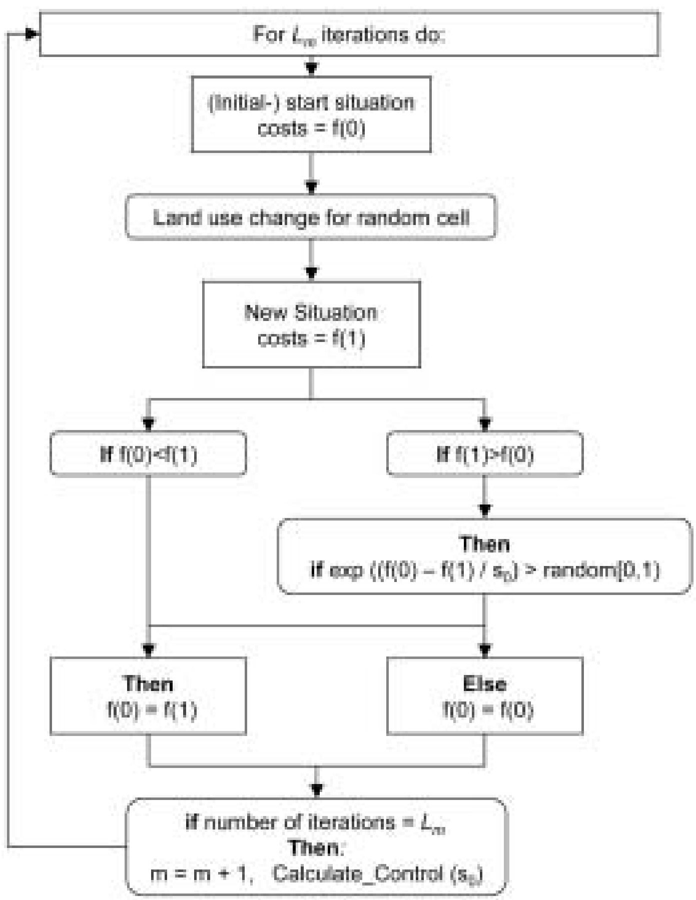

Figure 3. Flow diagram of the simulated annealing algorithm.

Within the current study, parameter settings developed by Aerts & Heuvelink (2002) were used. The start value of the freezing parameter was chosen such that within 500 trial iterations, 80% of all calculated costs were greater than the original situation. The decrease parameter r was set to 0.85 and the iteration length L per temperature stage was set to 1000.

A Genetic Algorithm (GA)

Genetic algorithms are based on the evolution theory in which the selection of new generations is based on the fittest species. There are many forms of such algorithms, but in this research the GA is defined by three stages. It starts with (a) the generation of the initial population of M_0 'parent solutions' (i.e. of land-use maps). This as opposed to SA, that uses the current land-use map as initial situation. Next, (b) the algorithm randomly selects pairs of 'parent' solutions, and a (c) 'crossover' procedure generates M_1 'child solutions' (read: new land-use maps). Then, (d) these child solutions are 'mutated'. The best M_0 of the $M_0 + M_1$ solutions are retained to form the next parent population until specified convergence criteria are met (Stewart *et al.*, 2002). The three stages are briefly discussed.

Random generation of parent solutions. For each solution, an initial cell is chosen at random. Next, a land use is allocated randomly for the selected cell with a probability for each land use set proportional to a *selection value* δ_{ijk}. A maximum value of $\delta_{ijk} = 1$ indicates the cheapest allocation of land use k to this particular cell. δ_{ijk} is set to 0, where constraints prohibit allocation of land use k to cell (i, j). Once a land use has been allocated to a first cell, an attempt is made to expand this into a cluster of reset minimum number of cells with this land use k. This is achieved by randomly selecting cells, which are neighbors to the currently evolving cluster.

Selection of parents. There is a need to select preferentially for the fittest parents, so that the probability of choosing a particular solution for a crossover pairing should be an increasing function of fitness (i.e. a decreasing function of the scalarizing function). The solution with the smallest value of the scalarizing function is allocated a relative probability of 1, and that with the largest value a relative probability specified by a parameter ε $(0 < \varepsilon < 1)$. Relative probabilities of selection for the remaining elements of the parent population are linearly interpolated between ε and 1.

Definition of crossover. The major problem relates to the crossover process from a pair of parent solutions, which uses two parent solutions to form a new child solution. Conventionally, genetic algorithms tend to perform a crossover by taking half the solution from one 'parent' and the other half from the other. Applied to the context here, this means that if each cell is independently allocated to one of the parent uses by random selection, the resulting child map will tend to be highly fragmented, leading to much worse performance on the spatial criteria than for either of the parents. On the other hand, simply splitting the region into two equal areas, and applying the solution from one parent to the one area, and from the other parent for the other is also not a good option.

It is proposed to use an alternative approach to the crossover process. For those cells, which are allocated to the same land use in both parent solutions, this same land use applies (naturally) to the child solution. Then for each pair of different land uses, say k and l, such that $k < l$, all cells are identified such that the land use is k in one parent solution and l in the other.

Definition of mutation. After crossover, a mutation is applied by random selection of a block D of cells consisting of R_D rows and C_D columns (where R_D and C_D are tuning parameters). The land uses from this block are deleted, and replaced by applying the same random selection algorithm as used for generating the initial population.

Testing the Model

Introduction

In order to demonstrate the goal-programming approach for its capacity to allocate compact patches of land use (and possibly connectivity), a fictive study area of 20 x 20 cells was developed with four land uses. The initial distribution of these land uses is presented in Figure 4 (right). Following the model formulations of earlier sections of the paper, five objectives are distinguished, which either can be categorized in additive cost objectives or objectives that relate to maximizing compactness of land use of the same type. The objectives are to:

(1) Minimize cost to allocate land use.
(2) Minimize cost of changing land-use types (transition costs).
(3) Minimize fragmentation (Spatial objective A).
(4) Maximize the largest cluster (Spatial objective B).
(5) Maximize overall compactness (Spatial objective C).

The values related to objective 1 are defined as C_{ijpk}, summed over all cells in the area, for each land-use type k (for k = 1, …, 4). These values are represented by cost maps, which either have a uniform or a variable value. All values are scaled between 1 and 10. The cost maps per land-use type are presented in Figure 5 where low costs mean a value of 1 and high costs refer to a value of 10.

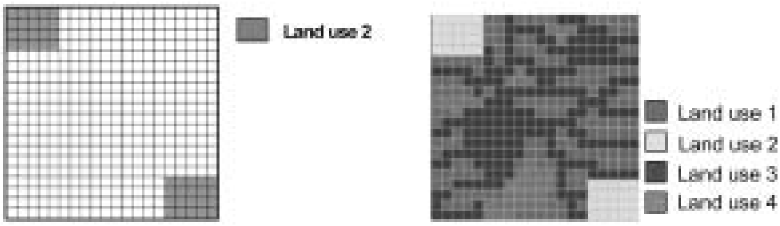

Figure 4. Land-use map showing the areas with fixed land use type 2 (left) and the current land use map (*right*).

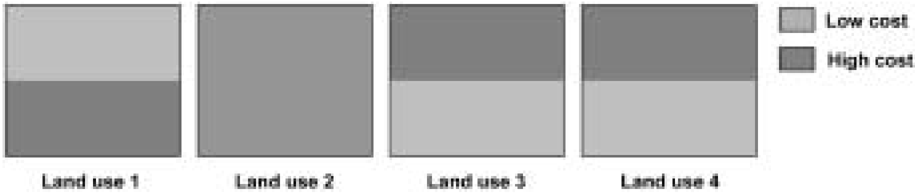

Figure 5. Cost maps for allocating land use k to a cell (for $k = 1, \ldots, 4$).

Table 1. Transition matrix, showing costs (Euro/cell) to change current land use k_c *into* future land use k

	Current land use k_c			
Future land use type (k_f)	**L. use 1**	**L. use 2**	**L. use 3**	**L. use 4**
Land use 1	0	10	9	8
Land use 2	10	0	10	9
Land use 3	9	10	0	10
Land use 4	8	9	10	0

Furthermore, transition costs for changing current land use k_c into future land use k_f are presented in Table 1. Management costs for maintaining certain land-use types are not considered in this model, but can easily be integrated following the same approach as with objective 1.

Constraints

As described in the section describing the goal-programming model, the following three constraints were used for the case study:

- A maximum and minimum required number of cells for each land-use type (lower and upper bounds) within the total area (Table 2).
- A minimum cluster size for each land-use type (Table 2).
- Pre-defined cells with a fixed land-use type (Figure 4, left).

The constraint values for lower and upper bounds and minimum cluster sizes are listed in Table 2. This Table also presents the current number of cells allocated to a specific land use k.

Parameter Settings

Four optimization runs were selected and solved using both SA and GA. Runs 1, 2 and 3 have their specific *goal* settings to test the model on its efficacy for generating compactness (Table 3). It should be noted that by increasing a goal value, the algorithm puts more effort in finding a solution that satisfies this value. For Run 3, the goal value for objective 3 'minimizing fragmentation' was increased to see

Table 2. Various spatial constraints for each land use type *k*

Land use type *k*	**Lower bound (cells)**	**Upper bound (cells)**	**Current (cells)**	**Min.cluster size (cells)**
Land use 1	90	120	142	1
Land use 2	120	150	40	1
Land use 3	80	120	182	1
Land use 4	70	100	36	1

Table 3. Goal value sets per objective, for each model run 1, 2, 3 and 4

	Objective values per model run				
Run no.	**Obj. 1 Min alloc. cost**	**Obj. 2 Min trans. cost**	**Obj. 3 Min. fragm.**	**Obj. 4 Max cluster size**	**Obj. 5 Max comp.**
Run 1	0.5	0.5	0.5	0.5	0.5
Run 2	0.5	0.5	1.0	0.5	1.0
Run 3	0.5	0.5	1.0	0.5	0.5
Run 4	0.5	0.5	1.0	0.5	0.5

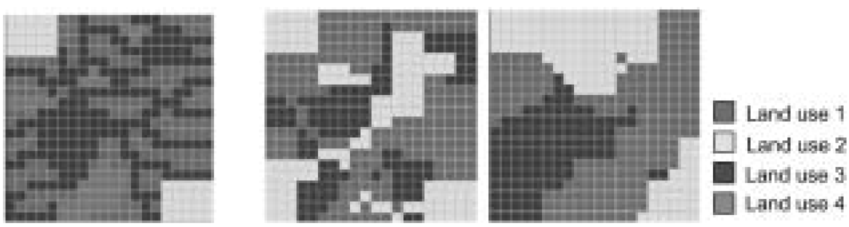

Figure 6. Initial land-use map (left), optimized maps using simulated annealing (centre) and the genetic algorithm (right).

whether connectivity between patches of land use would increase. In addition, connectivity of patches for land use 2 has been forced by using a new cost map for land use 2 but using the same goal value setting as in Run 3. All other model settings were kept constant across all optimization runs.

Results

Run 1 evaluates compactness according to a standard parameter setting where cost objectives and spatial compactness objectives are equally preferred. Figure 6 shows the optimization results. The map on the left shows the initial situation. The map in the centre shows the final land-use allocation using SA and the map to the right shows the final situation achieved using GA. For the GA result, it can be seen that land use 1 is allocated in the upper half of the area, since this is indeed the cheapest area shown in Figure 5. Land uses 3 and 4 compete for the other half of the area, as this is the cheapest area for both these land uses. Land use 2 is allocated somewhat scattered over the area in the solution generated by the SA procedure. This is confirmed by the objective values for objective 3 (minimize fragmentation) and objective 5 (maximize compactness), which both score worse for the SA result as compared to the GA result (Table 4).

Within run 2, the goal values for the spatial objectives 'minimizing fragmentation' and 'maximizing compactness' were increased, in order to achieve a more compact land-use pattern. Figure 7 shows the results for using SA and GA. Both optimization results have indeed improved the compactness of the land-use patterns. However, visual inspection shows that GA performs again somewhat better than SA. The objectives value for spatial objective 4 is better for SA than for GA. However, the GA values for objectives 3 and 5 are better than those for SA. Table 4 shows that the objective values are 9 against 5 and 29.31 against 21.81, when comparing SA against GA on objectives 3 and 5 respectively. Hence, it can be concluded that in terms of overall compactness, GA performs better than SA. Transition costs are slightly higher in the GA result, but allocation costs are kept lower in the GA run 2, as compared to the SA run 2.

Within run 3, the effectiveness of the objective 'minimize fragmentation' is evaluated for determining how the two fixed clusters of land use 2 could be *connected* at the lowest cost. In this respect, it should be noted that the cost for allocating land use 2 is equal for all cells (see Figure 5). Therefore, finding a 'least cost path', a term often used in geographic research, between the two fixed areas, can be forced by only

Table 4. Model results: objective values for all runs 1 to 4, for both SA and GA

Run no.	Obj. 1 Min. alloc. cost	Obj. 2 Min. trans. cost	Obj. 3 Min. fragm.	Obj. 4 Max. cluster size	Obj. 5 Max. comp.
Simulated annealing					
Run 1	1600	2022	15	2.68	29.22
Run 2	1220	1873	9	2.21	29.31
Run 3	1960	1979	4	4.00	36.31
Run 4	380	2047	11	2.47	27.32
Genetic algorithm					
Run 1	1040	1761	8	3.40	22.13
Run 2	922	1932	5	3.82	21.81
Run 3	800	2074	4	4.00	29.44
Run 4	650	2256	17	2.83	31.29

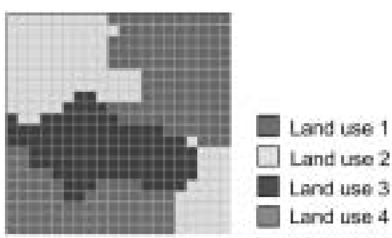

Figure 7. Results for run 2, using increased goal values for 'minimizing fragmentation' and 'maximizing compactness'. The map shows the result using the genetic algorithm.

increasing the goal value for spatial objective 3 (minimizing fragmentation). It is expected that the model will connect the two separate areas, since it is cheaper to 'de-fragment' areas with the same land use.

For run 3, the goal value for spatial objective 3 is set to twice the goal values for the other objectives. The results for using both SA and GA are presented in Figure 8 (left and right, respectively). It is shown that the overall compactness is better for the GA than for SA (21.44 against 36.31), but that both SA and GA connect the two areas for land use 2 that were initially situated separately. However, the two algorithms have found a different route. SA indicates an optimal route through the centre of the area, and GA a route along the right and upper borders. The SA result, however, is more expensive as compared to the GA run with 1960 against 800.

For the final run 4, a new cost map was prepared for land use 2 (Figure 9), with two 'cheaper channels' connecting the two fixed isles of land use 2. As in run 3, the

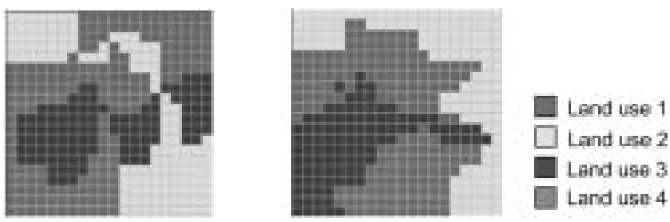

Figure 8. Results for run 3, using an increased goal values for minimizing fragmentation. The map on the left is the final result using simulated annealing and the map on the right is the final result using the genetic algorithm.

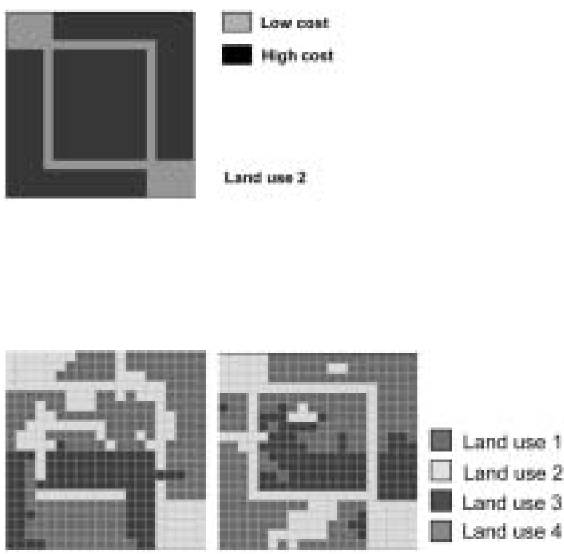

Figure 9. Results for run 4, using an increased goal value for minimizing fragmentation and a new cost map (top) for land use 2. The map on the left is the final result using simulated annealing and the map on the right is the final result using the genetic algorithm.

goal value for the objective 'minimize fragmentation' was again set twice as high as the goal for the other objectives. It can be seen that both SA and GA produced a result that uses the cheaper channel as a path for connecting the two separate land use 2 areas. The GA run produced nice connections using the preset channels. However, in terms of the values for objectives 3 and 5, namely minimization of fragmentation and maximization of compactness the GA results are worse as compared to the SA results. In addition, the costs for the SA run 4 are lower, as opposed to the GA results.

In general, the SA algorithm appears to be slower than GA for all runs. The overall solution time for solving the 20 x 20 map was about 30 seconds for GA and about 90 seconds for the SA procedure.

Jisperveld Case Study

Introduction

Jisperveld is the largest connected brackish fen-meadow area of Western Europe. It is situated in the Northwest of the Netherlands and measures about 2000 ha. It is a typical Dutch landscape with drained peat meadows in polders below sea level. The whole area is criss-crossed with water, which gives it its special character (Figure 10). The high natural value of the area comes from the presence of rare meadow birds and the existence of special vegetation that both rely on wet conditions.

The Jisperveld area is subject to a debate on how to both plan and manage the area in the future. It appears that governmental planning policy for land use is changing from predominantly agriculture to a combined agriculture and nature area. This can be achieved by a change in water levels, which are fully controlled by the regional water board. For example, a higher water table in the meadow area attracts rare birds but lowers agricultural use.

A process of discussion and negotiation with stakeholders and institutions in the area has already begun. Different stakeholders, such as agricultural organizations, recreational organizations, nature conservation organizations and regional autho-

Figure 10. Aerial photo of Jisperveld.

rities, each have their own ideas about the future land use. For this process, it is proposed to support planners with a design tool, such as the one described in this paper, that allows for searching new land-use plans in a participatory approach with (local) policy makers and stakeholders.

Model Set-up

In order to apply the goal-programming approach, the current land-use map of the area was simplified and the size reduced into a map of 400 ha. The number of land-use types was reduced from 33 to nine. Figure 11 presents the map of the current situation with nine land-use types.

The main goal of this case study is to search for a nature recreation plan that allows for both more recreation opportunities and higher environmental values. For this, the plan should contain the allocation of two new land-use types: 'extensive agriculture' and 'water (limited access)', both of which are not yet present. The emphasis is on generating compact patches of these two new land uses.

Following the model formulations, six objectives are distinguished:

(1) Maximize the natural value of the area.
(2) Maximize the recreational value of the area.
(3) Minimize cost of changing land use.
(4) Minimize number of clusters.
(5) Maximize cluster size.
(6) Maximize compactness.

It should be noted that both natural and recreational values can be seen as costs that contribute positively to the overall objective function. Furthermore, transition costs for changing current land use k_c into future land use k_f are presented in Table 5. Management costs for maintaining certain land-use types are not considered in this model, but can be easily integrated.

Generating Land-use Plans

Within the first step of using the optimization software, the user is asked to set goals for each of the above mentioned objectives. Next, an optimization procedure (either

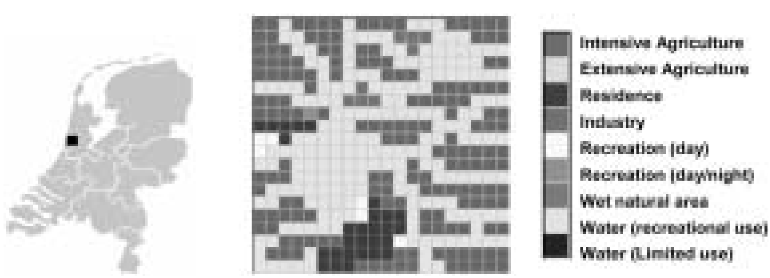

Figure 11. Location of the Jisperveld area in the Netherlands, indicated with the black dot (left) and the simplified current land-use map of the Jisperveld, measuring 20 x 20 cells (right).

Table 5. Transition matrix, showing costs (Euro/cell) to change current land use k_c into future land use k_f

	Current land use k_c								
Future land use type (k_f)	**Int. agri.**	**Ext. agri.**	**Residence**	**Industry**	**Recr. (day trip.)**	**Recr. (over night)**	**Wet nature**	**Water (recr)**	**Water (limit)**
Intensive agriculture	0	1000	10000	500	–	7000	–	–	–
Extensive agriculture	–	0	–	–	–	–	–	–	–
Residence	–	–	0	–	–	–	–	–	–
Industry	–	–	–	0	–	–	–	–	–
Recreation (day trippers)	–	–	9000	–	0	5000	–	–	–
Recreation (overnight)	–	–	–	–	–	0	–	–	–
Wet natural area	–	–	–	–	–	–	0	–	–
Water (recreational use)	–	–	–	–	–	–	–	0	1000
Water (limited access)	–	–	–	–	–	–	–	–	0

using simulated annealing or genetic algorithm) tries to meet these goals as much as possible. Three goal settings have been defined in which the goals related to each of the spatial objectives were varied while the goals for the additive cost objectives were kept constant. For each goal setting, the model was solved using both simulated annealing and the genetic algorithm. As stated in earlier sections, increasing a goal value attached to an objective, while keeping others constant will put more emphasis on that particular objective.

Table 6 shows the values of the goals used in three goal settings (Run 1, Run 2 and Run 3). These values are standardized between 0 and 1. The additive goal values for the objectives nature, recreation and costs are set to 0.5 of the maximum value (maximum value = 1). In Run 1 the goal value related to 'number of clusters' is set to 0.9 and the other spatial goal values are set to 0.4. In Run 2 the 'cluster size' goal is set to 0.9 of the maximum value and in Run 3 the compactness goal is set to 0.9 of the ideal maximum.

Figure 12 shows the results for running each of the three goal settings. On the left, three land-use plans (SA1, SA2 and SA3) are presented that were generated by simulated annealing. On the right, the Figure shows three land-use plans (GA1, GA2 and GA3) generated by the genetic algorithm.

Visual inspection of maps SA1 and GA1 compared with the other maps shows that both algorithms do indeed produce the least fragmented plans for SA1 and GA1. This can be seen best for the land-use type 'water limited use'. Furthermore, when comparing GA1 and GA2, it shows that the cluster size of land use 'water limited use' in GA2 is larger than in GA1, as would be expected. However, this is not the case when comparing SA1 and SA2, as SA2 contains three separate clusters as opposed to two clusters in SA1. Hence, the simulated annealing algorithm performs less well when increasing the goals for the objective 'maximize cluster size'. In addition, when comparing SA3 and GA3, it appears that the genetic algorithm produces more compact patches of the new land use 'water limited use' and 'extensive agriculture'.

In general, the results show that the resulting plans GA1, GA2 and GA3 are either intensive or extensive agriculture. The resulting plans SA1, SA2 and SA3, however, often mix these two types of land use. This can be explained in that both approaches may find local optima rather than the overall optimal solution.

The visual inspections of the model results are supported by the objectives values that are given for each map result in Table 7. For the current situation (second row of the Table) the 'number of clusters' objective achieves 0.44 of the ideal value

Table 6. Three model sets with different settings for the goal values relative to the ideal objective '1'

	Nature	Recreat.	Trans. cost	No. clusters	Cluster size	Compact
Range	0–1	0–1	0–1	0–1	0–1	0–1
Run 1 No. clust	0.5	0.5	0.5	0.9	0.4	0.4
Run 2 Cluster size	0.5	0.5	0.5	0.4	0.9	0.4
Run 3 Compact	0.5	0.5	0.5	0.4	0.4	0.9

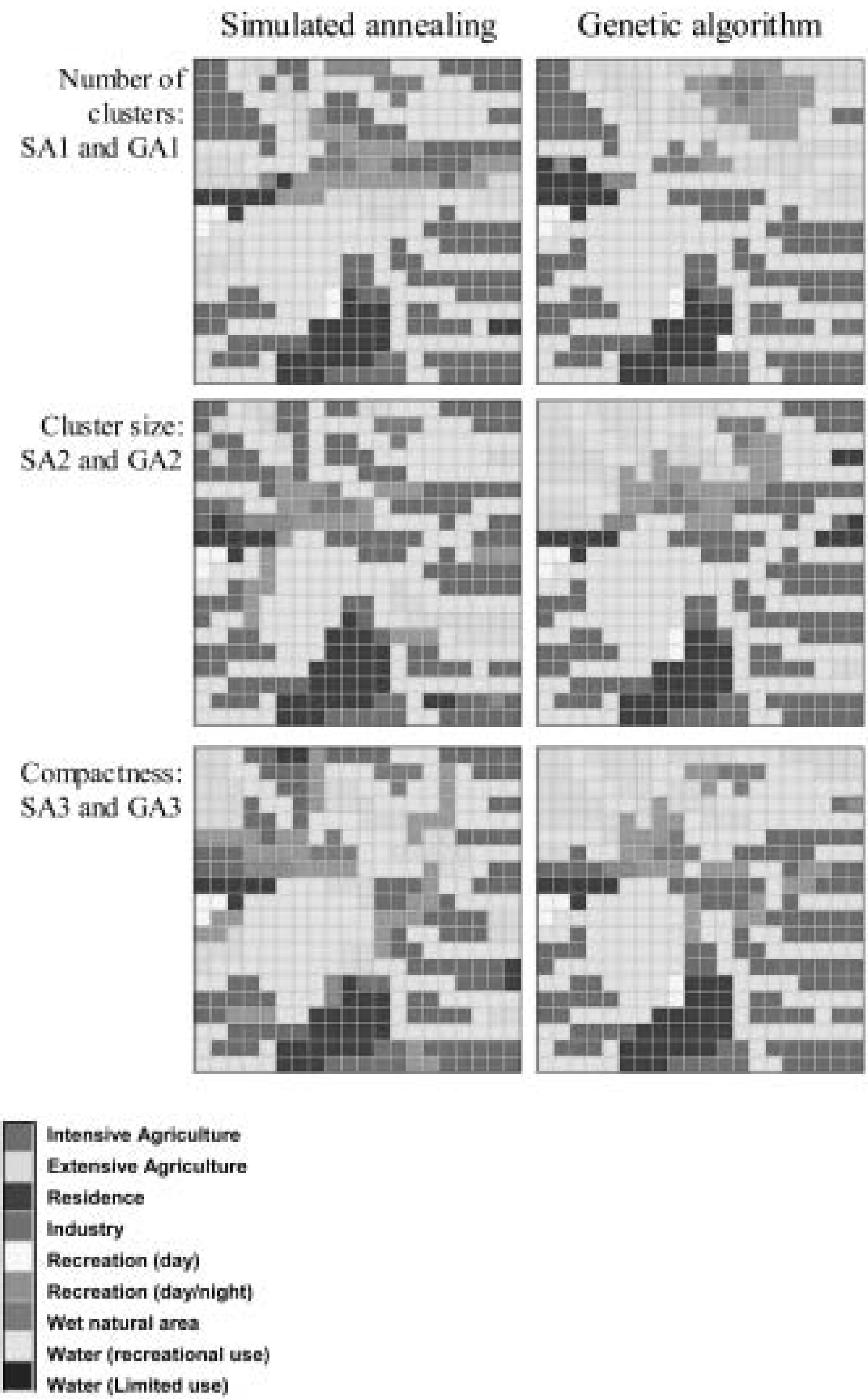

Figure 12. Model results using three goal settings. On the left, the results using the simulated annealing algorithm. On the right, the results using the genetic algorithm.

Table 7. Model results according to the three sets of goals for the spatial objectives solved using both simulated annealing and genetic algorithm

	Total value	**Nature**	**Recr.**	**Costs**	**No. clusters**	**Cluster size**	**Comp.**
Range	0 – 1	0 – 1	0 – 1	0 – 1	0 – 1	0 – 1	0 – 1
Current	0.00	0.00	0.54	1.00	0.44	0.71	0.68
Simulated annealing							
SA1	0.9389	0.18	0.16	0.44	0.36	0.49	0.56
SA2	0.9870	0.11	0.17	0.31	0.17	0.59	0.55
SA3	0.9947	0.20	0.06	0.26	0.00	0.40	0.76
Genetic algorithm							
GA1	0.9711	0.38	0.33	0.35	0.36	0.47	0.53
GA2	0.9912	0.25	0.18	0.30	0.33	0.57	0.58
GA3	0.9971	0.27	0.24	0.43	0.03	0.39	0.74

('1'), the 'cluster size' objective achieves 0.71 of the ideal and the compactness objective achieves 0.68. The resulting plans of the three sets in Table 7 clearly show higher achievements for the objectives for which goals were set to 0.9. This applies for both the simulated annealing results as for the genetic algorithm results. The achievement of the 'number of clusters' objective in SA1, for example, is 0.36 against 0.17 and 0.0 achievement of 'the number of clusters' in SA2 and SA3, respectively. The Table also shows that the genetic algorithm generates better results than simulated annealing. This can be seen from the total objective value and especially applies to the nature and recreation objectives. On the other hand, the achievement of the goals for the cluster size objective is somewhat better for the simulated annealing results.

Conclusions and Discussion

The first objective of this study was to investigate whether goal programming (GP) combined with simulated annealing (SA) and genetic algorithms (GA), is an attractive alternative for designing spatial resource allocation alternatives. For this, a generalized GP approach has been used, based on Stewart *et al*. (2002) to solve a MLUA problem. It is thought that GP has advantages over, for example, Multi-criteria analysis since it allows land-use planners to design a plan using preset goals. In contrast, many planners have difficulties setting values or weighting the relevant attributes involved in the multi-criteria analysis, especially when the alternatives are not at hand (Aerts & Heuvelink, 2002).

A second objective was to develop three spatial objectives, based on commonly used compactness characteristics that address size, perimeter and area of a cluster of the same land use. The compactness objectives refer to minimizing fragmentation, maximizing the largest cluster of a land-use type and maximizing overall compactness. The user may vary the goal for each of the spatial objectives, thereby generating different land use designs.

A third objective was to test the GP model by solving it using either SA or GA. First, the model was solved in four test runs using both the SA and GA algorithms.

The test area was a fictive area of 20 x 20 cells with simplified cost fields. Each run was prepared with different sets of compactness goals. The model produced compact patches of land use using both SA and GA, although the GA results were somewhat better than the SA results in terms of compactness. When increasing the goal value of the objective 'minimize fragmentation', two fixed separate areas of land use 2 were connected in the results for both SA and GA.

The final objective was to apply both algorithms to a case study in the Jisperveld area in The Netherlands. The aim was to generate a nature recreation plan by allocating two new land uses. By varying the input, a set of development plans was generated that gives an overview of possible plans for the study area. Again, both algorithms generated compact patches of new land use, although GA performed somewhat better than SA in terms of reaching compactness goals. In addition, the SA algorithm was slower than GA.

The performance of spatial objective 3 ('minimize fragmentation') is interesting for promoting connectivity. Research has demonstrated that spatial connectivity objectives appear to be difficult to model and solve through its highly non-linear formulations (see e.g. Cova, 1999; Cova & Church, 2000a, 2000b). However, it was demonstrated that connectivity can be achieved using the compactness objective 'minimize fragmentation'.

Land-use allocation problems are often complex as they involve multiple stakeholders with conflicting goals and objectives (O'Connell & Keller, 2002). The methodology presented in this paper can be applied to support such a complex multi-stakeholder process since it allows for trading-off different objectives in the planning process.

References

Aarts, E. & Korst, J. (1989) *Simulated Annealing and Boltzman Machines. A Stochastic Approach to Combinatorial Optimization and Neural Computing* (New York: John Wiley).

Aerts, J. C. J. H. (2002) Spatial decision support for resource allocation, integration of optimization, uncertainty analysis and visualization techniques, PhD Thesis, University of Amsterdam (Amsterdam: Thela Thesis publishers).

Aerts, J. C. J. H. & Heuvelink, G. B. M. (2002) Using Simulating annealing for resource allocation, *International Journal of Geographic Information Science*, 16, pp. 571–587

Aerts, J. C. J. H., Eisinger, E., Heuvelink, G. & Stewart, T. J. (2003) Multi-site land use allocation for spatial decision support using integer programming, *Geographical Analysis*, 35, pp. 512–534.

Brookes, C. J. (1997) A parameterized region-growing program for site allocation on raster suitability maps, *International Journal of Geographical Information Science*, 11, pp. 375–396.

Boston, K. & Bettinger, P. (1999) An analysis of Monte Carlo integer programming, Simulated annealing, and Tabu search heuristics for solving spatial harvest scheduling problems, *Forest Science*, 45, pp. 292–301.

Church, R. L. (2002) Geographical Information Systems and location science, *Computers and Operations Research*, 29, pp. 541–562.

Church, R. L., Stoms, D. M. & Davis, F. W. (1996) Reserve selection as a maximal covering allocation problem, *Biological Conservation*, 76, pp. 105–112.

Cova, T. J. (1999) A general framework for optimal site search, PhD Thesis, University of California, Santa Barbara.

Cova, T. J. & Church, R. L. (2000a) Contiguity constraints for single-region site search problems, *Geographical Analysis*, 32, pp. 306–329.

Cova, T. J. & Church, R. L. (2000b) Exploratory spatial optimization in site search: a neighborhood operator approach, *Computers Environment and Urban Systems*, 24, pp. 401–419.

Diamond, J. T. & Wright, J. R. (1989) Efficient land allocation, *Journal of Urban Planning and Development*, 115, pp. 81 – 96.

Fonseca, C. M. & Fleming, P. J. (1995) An overview of evolutionary algorithms in multi objective optimization, *Evolutionary Computation*, 3, pp. 1 – 16.

Greenberg, H. J. (2002) *Mathematical Programming Glossary*. Available at http://www.cudenver.edu/~hgreenbe/glossary/glossary.html

Jaszkiewicz, A. (2002) Genetic local search for multi-objective combinatorial optimization, *European Journal of Operations Research*, 137, pp. 50 – 71.

Jones, G. J. Meneghin, B. J. & Kirby, M. W. (1991) Formulating adjacency constraints in linear optimization models for scheduling projects in tactical planning, *Forest Science*, 37, pp.1283 – 1297.

Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983) Optimization by Simulated annealing, *Science*, 220, pp. 671 – 680.

Laarhoven, P. J. M. (1987) Theoretical and computational aspects of Simulated annealing, PHD Thesis, Erasmus University, Rotterdam.

Levine, B. (2002) Presentation for EE652. Available at http://microsys6.engr.utk.edu/~levine/EE652/Slides/tsld001.htm

Lockwood, C. & Moore, T. (1993) Harvest scheduling with spatial constraints: a Simulated annealing approach, *Canadian Journal of Forest Research*, 23, pp. 468 – 478.

Malczewski, J. (1999) *GIS and Multicriteria Decision Analysis* (New York: John Wiley).

Murray, A. T. & Church, R. L. (1995) Measuring the efficacy of adjacency constraint structure in forest planning models, *Canadian Journal of Forest Research*, 25, pp. 1416 – 1424.

O'Connell, I. & Keller, C. P. (2002) Design of decision support for stakeholder driven collaborative land valuation, *Environment and Planning B*, 29, pp. 607 – 628.

Ridgley, M., Penn, D. C. & Tran, L. (1997) Multicriterion decision support for a conflict over stream diversion and land-water reallocation in Hawaii, *Applied Mathematics and Computation*, 83, pp. 153 – 172.

Steuer, R. (1986) The Tchebycheff procedure of interactive multiple objective programming, in: B. Karpak & S. Zionts (Eds) *Multiple Objective Decision Making and Risk Analysis Using Micro Computers* (Berlin: Springer-Verlag).

Stewart, T. J. (1991) A multi criteria decision support system for R&D project selection, *Journal of the Operational Research Society*, 42, pp.1369 – 1389.

Stewart, T. J., Herwijnen, M. & Janssen, J. (2002) *A Genetic Algorithm Approach to Multi Objective Land Use Planning* (Under review).

Sundermann, E. (1995) PET Image reconstruction using Simulated annealing, in: *Proceedings of the SPIE Medical Imaging 1995 Conference, San Diego, Image Processing*, pp. 378 – 386.

White, R. & Engelen, G. (1997) Cellular automata as the basis of integrated dynamic regional modelling, *Environment and Planning B: Planning and Design*, 24, pp. 235 – 246.

Wierzbicki, A. P. (1999) Reference point approaches, in: T. Gal, T. J. Stewart & T. Hanne (Eds) *Multi Criteria Decision Making: Advances in MCDM Models, Algorithms, Theory and Applications* (Boston: Kluwer Academic Publishers).

Williams, J. C. & Revelle, C. S. (1998) Reserve assemblage of critical areas: a zero-one programming approach, *European Journal of Operations Research*, 104, pp. 497 – 509.

Wright, J., Revelle, C. S. & Cohon, J. (1983) A multi-objective integer programming model for the land acquisition problem, *Regional Science and Urban Economics*, 12, pp. 31 – 53.